Charles, Ada, and the engine

**Difference Engine**

Every great invention needs a few iterations for it to show its true potential, and every great story needs a great beginning. So, for this story we must go back to 1819, when 28-year-old mathematician Charles Babbage began constructing a miniature version of a machine he believed would revolutionize the field of mathematics. This machine named the “difference” engine, was able to tabulate logarithms and trigonometric functions and show the results via the arrangement of the mechanical dials on the machine. Three years later, in 1822, he announced his invention to the Royal Astronomical Society. The British government took immediate interest in his invention and in 1823 gave Babbage a grant to build a full-scale version of the machine. Work on this machine consumed the next 10 year of Babbage’s life, but in the end was abandoned due the expense of metalworking at the time. By the time that British government gave up on the project in 1833, Babbage, who now had 10 years of mechanical engineering under his belt had already began thinking of how to create a machine that could be used for more generalized computation.

**Beginning of Analytical engine**

Babbage’s design of the Differential Engine was not without its drawbacks, the machine was unable to perform any other operation other than addition, and whenever a new constant was needed, the value would have to manually input into the machine. Babbage realized the disadvantages to the Differential Machine and began work on the Analytical machine around September, 1834 in which he spent the rest of his lifetime working on improving, along with many other accomplishments. Although his machine was never built, the notes of his design pioneered many ideas of modern computing including some kind of programmable input, a printed output, memory that all values needed for computations, and a central processing unit that performed the arithmetic of the inputted algorithm.

**How the Analytical engine worked**

Here is artist Sydney Padua’s rendition of what the Analytical machine would have looked like, had it ever been built. This large machine may be daunting to look at, but if we inspect the parts of this machine, we can see that this machine is structured similarly to a modern computer. Let’s start off with number 1 over here, which is the memory of the engine. This part of the machine would store variables for use in future computations. 2, This big round piece would be the CPU or the “brain” of the machine able to do computations and execute instructions given to it. 3 is the power of the machine, which is a steam engine. 4 is a printer that would have the ability to print out the value of whatever it was computing. 5, 6, and 7 are all various “punch cards” that make up the algorithm of the program. The punch cards have holes punched out of them in very specific patterns that the machine can then read and understand what instruction to execute based on the pattern in which the holes were punched. There were to be 3 separate slots for reading in punch cards that were all used for different purposes. One slot read in the instructions that need to be executed, another one reads in various numbers that the machine could use arithmetic computation, and the final one reads in what computed variables need to be stored and where to store them in the memory. And finally, 8, which are the barrel controllers. These barrel controllers stored “microprograms” on them that would tell the machine how to add, subtract, multiply, and divide.

**Ada Lovelace**

Babbage’s notes on the analytical engine had many other mathematicians interested, including Luigi Menabrae and Ada Lovelace. In 1841 mathematician Luigi Menabrae wrote an article briefly describing Babbage’s analytical machine and how it could potentially be used in arithmetic calculations. Ada Lovelace, whom Babbage already had a person acquaintanceship with, was asked to translate Luigi’s article from French into English, and to add her own notes on the topic began doing so in 1842. After nine months, Ada had translated Luigi’s article, added her own notes, and included instructions for the machine which is now widely regarded as the world’s first computer algorithm.

**Bernoulli**

The algorithm that Ada included cannot be fully understood without taking a side step for a moment, to a man named Jakob Bernoulli. Bernoulli was working to solve the problem finding a general form of the summation of a value when raised to power p. Before Bernoulli, another mathematician named Johann Faulhaber had already discovered general formulas for this for powers of k up to 17. By manipulating these formulas, Bernoulli was able to notice a reoccurring pattern of numbers that was being multiplied in each of the formulas. If we simplify all of this craziness, we can see that this formula is simply two terms being multiplied together, and being summed all up for all terms up to power k. B is part of a sequence of numbers now known as the Bernoulli numbers. The first eight numbers of this sequence are 1, ±1/2, 1/6, 0, -1/30, 0, 1/42, 0, and -1/30. The other variable here, A, can be computed using Pascal’s triangle. This was a huge advance in mathematics, but even with Bernoulli’s method, this is still very computationally cumbersome, that’s where Lovelace’s algorithm comes in.

**How the algorithm works**

So now we can go back to Lovelace’s algorithm, which computes the 8th number of the Bernoulli sequence, or a variable she had labeled as “*B*7”. Her program, in essence, was solving the following equation: *B*7=−1(*A*0+*B*1*A*1+*B*2*A*2+*B*3*A*3+*B*5*A*5+*B*6*A*6). We know that both *B*2 and *B*6 are both 0, we really just have *B*7=−1(*A*0+*B*1*A*1+*B*3*A*3+*B*5*A*5). Let’s get a look at some C++ code that I have written that executes Lovelace’s algorithm. We start off by declaring variables v1, v2, and v3 and initiating them to the values provided in the algorithm with v3 representing the value of n. The algorithm also uses variables V4 – V13 for computation, so we need to declare them too and just initialize them to zero. Variables v21-V24 represent the numbers of the Bernoulli sequence with V24 being the number we are attempting to calculate. Since the Bernoulli numbers show up in every case, and they are already known, we can just store them as constant values. If we follow the math of the algorithm, we can see that steps 1-7 compute the equation A0 = -(1 / 2) \* ((2n - 1) / (2n + 1)). Steps 9 and 10 calculate *B*1*A*1 by first calculating A1 = 2n / 2 in step 9 and then multiplying by the B1 in step 10. Steps 11 and 12 just simply add together what we have so far which is A0 + B1A1 and move our counter variable down. We then enter into a loop that repeats steps 13-23 until our counter variable has hit zero, as indicated in her notes. Also indicated in her notes, steps 13-16 and steps 17-20 are the exact same, so we can simply enter into another small loop that performs steps 13-16 twice. We can do this because the formulas for A3 and A5 follow a pattern. The equation for A3 is A3= (2n) (2n-1) (2n-2)/ 2 \* 3 \* 4 and the equation for A5 is A5= (2n) (2n-1) (2n-2) (2n-3) (2n-4) / 2 \* 3 \* 4 \* 5 \* 6. These equations are exactly what we are computing in steps 13-20. Step 21 multiplies A3 and A5 with its respective Bernoulli number to come up with *B*3*A*3 and *B*5*A*5. Step 22 calculates up our sum, which is either *A*0+*B*1*A*1+*B*3*A*3 or *A*0+*B*1*A*1+*B*3*A*3+*B*5*A*5 depending on where the program is in the loop. Step 23 moves our counter variable down. Step 24 happens once we exit the loop, and simply stores our final result into variable V24. The last step, number 25 would move up our n value by one so that we can compute the next Bernoulli number, if the program were to continue on.

When we run the program, we get the result 0.03 which is incorrect, with the actual value of the 8Th Bernoulli number being -0.03. This is intentional, with comments from Lovelace in her notes about the Analytical machine sorting the proper sign to print. An interesting thing to note, since V24 just has the value of zero stored into it before step 24, if we change step 24 to being “v24 = v24 - v13” instead of “v24 = v13 + v24” the correct result will be computed.

**Conclusion**

Babbage’s son would go on to build the “mill” of the Analytical machine, which was the machines main processing unit. Unfortunately, though, due to the sheer complexity of the full scaled machine, to this date it has never been built. His notes inspired some of the most fundamental concepts of computer science. Ada Lovelace’s contribution to the Engine should not go without recognition either, and it isn’t. She is often regarded as the world’s “first programmer”.

<http://math.ucr.edu/~res/math153/s12/bernoulli-numbers.pdf>

<https://trans4mind.com/personal_development/mathematics/series/sumsBernoulliNumbers.htm>

<https://gist.github.com/sinclairtarget/ad18ac65d277e453da5f479d6ccfc20e>

<https://en.wikipedia.org/wiki/Bernoulli_number>

<https://www.youtube.com/watch?v=5rtKoKFGFSM>

https://sydneypadua.com/2dgoggles/the-marvellous-analytical-engine-how-it-works/